CLIPPEDIMAGE= JP402004288A

PAT-NO: JP402004288A

DOCUMENT-IDENTIFIER: JP 02004288 A

TITLE: IMAGE PROCESSOR AND METHOD FOR MANAGING ITS FONT FILE

PUBN-DATE: January 9, 1990

INVENTOR-INFORMATION:
NAME
YAMASHITA, TAKETOSHI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
FUJI XEROX CO LTD	N/A

APPL-NO: JP63153890

APPL-DATE: June 22, 1988

INT-CL (IPC): G09G001/02;G06F015/64 ;G09G005/36 ;H04N001/387

US-CL-CURRENT: 345/471

ABSTRACT:

PURPOSE: To easily check errors by providing a font file with at least one check sum, sum-checking font data based on the check sum and detecting the presence or absence of the data errors of the font file.

CONSTITUTION: The font file is provided with the check sum and sum-checking is performed based on the check sum so as to detect the presence or absence of the data errors of the font file. Thus, even when the font file is destroyed by operational errors, etc., it is easy to know places where the data errors occur. Therefore, error-checking can easily be performed. When the font file is loaded from memory devices, such as a hard disk, onto an image memory, sum-checking is performed. Thus, the data errors of the font file can be detected when filing or loading is performed.

⑲日本国特許庁(JP)　⑪特許出願公開

⑫公開特許公報(A)　平2−4288

⑤Int.Cl.⁵	識別記号	庁内整理番号
G 09 G 1/02		L 8121−5C
G 06 F 15/64	450	F 8419−5B
G 09 G 5/36		8839−5C
H 04 N 1/387		8839−5C

⑬公開　平成2年(1990)1月9日

審査請求　未請求　請求項の数 2　(全6頁)

⑭発明の名称　イメージ処理装置及びそのフオントフアイル管理方法

⑳特　願　昭63−153890
㉒出　願　昭63(1988)6月22日

㉗発明者　山下　武利　埼玉県岩槻市大字岩槻1275番地　富士ゼロツクス株式会社岩槻事業所内
㉑出願人　富士ゼロツクス株式会社　東京都港区赤坂3丁目3番5号
㉔代理人　弁理士　木村　高久

明　細　書

1．発明の名称

イメージ処理装置及びそのフォントファイル管理方法

2．特許請求の範囲

（1）イメージデータとフォントファイルを同一のイメージメモリ内に展開して処理するイメージ処理装置において、

少なくとも一つのチェックサムを設けたフォントファイルと、

前記チェックサムに基づいてフォントデータのサムチェックを行ない、フォントファイルのデータエラーの有無を検出する検出手段と、

を具えたことを特徴とするイメージ処理装置。

（2）イメージデータとフォントファイルを同一のイメージメモリ内に展開して処理するイメージ処理装置において、

前記イメージメモリ内でイメージデータとフォントデータの合成及び加工を行なう工程と、

前記フォントファイルに設けられたチェックサムに基づいて、フォントデータのサムチェックを実行する工程と、

を具え、前記サムチェックによりエラーが検出されたときは、イメージメモリから画像出力部へのイメージデータの送出を中断するようにしたことを特徴とするイメージ処理装置のフォントファイル管理方法。

3．発明の詳細な説明

（産業上の利用分野）

この発明は、入力したイメージデータと漢字等のフォントデータを合成・加工して出力するイメージ処理装置と、そのフォントファイル管理方法に関する。

（従来の技術）

第4図は、従来のこの種のイメージ処理装置の構成ブロック図である。第4図において、CPU（中央処理装置）3は、画像入力部1から入力したイメージデータを、イメージメモリ5のイメージ領域に蓄積すると共に、ハードディスク4か

ら読み取ったフォントファイルを、イメージメモリ5のフォント領域に蓄積する。そして、イメージメモリ5に蓄積されたイメージとフォントをメモリ内で合成・加工し、画像出力部2において出力させる。

（発明が解決しようとする課題）

ところで、上記イメージ処理装置では、イメージメモリ内にイメージとフォントファイルをデータごとに格納しているが、イメージ領域とフォント領域に境界を設けてしまうと、片方の領域がオーバーフローした時に他の領域を使用できなくなるので、各領域を仕切るための境界は設けられていない。このため、例えば、イメージメモリからフォントファイルをコピーして取り出す時に、領域の指定ミスによりフォントファイルの一部を破壊してしまうことがあった。この場合、破壊された部分はデータエラーとなり、ＣＲＴディスプレイ等に表示すると、画像乱れとなって表われる。しかし、従来は、データのエラー箇所を確認することができなかったため、エラー箇所を見つけるために、フォントファイルのデータを最初から見直さなければならず、エラーチェックが煩雑になるという問題点があった。

この発明は、上記問題点に鑑みなされたもので、フォントファイルのデータエラー箇所を検出し、エラーチェックを容易に行なうことができるイメージ処理装置及びそのフォントファイル管理方法を提供することを目的とする。

（課題を解決するための手段）

上記目的を達成するため、この発明に係るイメージ処理装置は、フォントファイルに少なくとも一つのチェックサムを設けると共に、このチェックサムに基づいてフォントデータのサムチェックを行ない、フォントファイルのデータエラーの有無を検出する検出手段を設けたことを特徴とする。

また、上記イメージ処理装置のフォントファイル管理方法は、イメージメモリ内でイメージデータとフォントデータの合成及び加工を行なう工程と、フォントファイルに設けられたチェックサムに基づいて、フォントデータのサムチェックを実行する工程とを具え、前記サムチェックによりエラーが検出されたときは、イメージメモリから画像出力部へのイメージデータの送出を中断するようにしたことを特徴とする。

（作用）

イメージメモリ内においてイメージとフォントの合成・加工を行なった後、フォントデータに設けたチェックサムを用いて、フォントデータのサムチェックを行なう。そして、サムチェック終了後、フォントデータの合計があらかじめ設定されたサム値に対して正常か否かを検出手段により判断する。ここでデータエラーが発生していなければ画像出力部へイメージデータを送出し、データエラーが検出されたときは、そのデータエリアに関するデータの画像出力部への送出を中断する。したがって、データエラーの発生した箇所を速やかに検出することができる。

（実施例）

以下、この発明の実施例を図面と共に詳細に説明する。

第２図は、この発明に係るイメージ処理装置の一実施例を示す構成ブロック図である。図において、１１はイメージメモリであり、イメージデータとフォントファイルが格納される。ここで用いられるフォントファイルの一例を第１図に示す。フォントファイルは、４０９６ワードを一単位とする複数のバンクから構成され、０バンク目にチェックサムテーブル、１～３バンク目にフォントコードを示すフォントコードテーブル、４～１１バンク目にフォントデータが配置されている。前記０バンク目のチェックサムテーブルのアドレス（１～１１）は、フォントファイルの各バンク番号１～１１に対応している。例えば、バンク３のチェックサム（ＳＵＭ－３）は、チェックサムテーブルのアドレス３に入っている。なお、バンク０に対応するアドレス０には、チェックサムは入っていない。また、チェックサムテーブルのアドレス１２～４０９５にはデータが入っていないものとする。また、前記イメージメモリ１１は、

上位１２ビットをバンク番号（０～１１）、下位１２ビットをバンク内アドレス（０～４０９５）として、４０９６×４０９６（ワード）の空間に分割されている。なお、１ワードは１６ビットで構成されている。１２はフォントファイルが記憶されたハードディスクであり、ハードディスクコントローラ１５によりデータの入・出力が制御される。１３は合成・加工されたイメージデータを出力するためのプリンタであり、その出力はプリンタコントローラ１６により制御される。なお、この例では画像出力部にプリンタを用いたが、出力装置としてはＣＲＴディスプレイやＸＹプロッタ等を用いてもよい。１４はイメージデータを入力するためのイメージリーダーであり、データの入力はイメージリーダーコントローラ１７により制御される。

１８はＲＯＭであり、後述するサムチェックの処理手順のフローチャートに基づいたプログラム等が記憶されている。１９はＲＡＭであり、サムチェックや出力を行なう前のイメージデータが格納される。２０は各部の制御を行なうＣＰＵであり、イメージ処理においてはイメージメモリ１１でのイメージとフォントの合成・加工を制御する。また、前記ＲＯＭ１８に記憶された処理プログラムに基づいて、フォントデータのサムチェックを実行する。ＣＰＵ２０でのサムチェックは、次式(1)の計算結果に基づいて行なわれる。

判定値＝

$$adr(0)+adr(1)+adr(2)+\cdots+adr(4095)+(SUM\text{-}j) \quad \cdots\cdots(1)$$

この例では、算出された判定値が０のときにデータエラーなしと判断する。すなわち、ｊバンク目の全データの合計と、そのバンクのサム値ＳＵＭ－ｊを加算して０であればフォントファイルが正常であると判断する。

２１はアドレスコントローラ（シーケンサ）であり、前記プリンタコントローラ１６、イメージリーダコントローラ１７等の入出力制御装置や、イメージメモリ１１とＣＰＵ２０との間のデータ転送に使用される。２２はデータレジスタであり、ＣＰＵバスとイメージバスとの間でＲＡＭ１９やイメージメモリ１１に格納されたバンクのアドレス番号を順次指定する。

また、２３はホストコンピュータ、２４は前記ホストコンピュータ２３とＣＰＵバスを結ぶホストＩ／Ｆ（インターフェス）である。

次に、上記イメージ処理装置におけるサムチェックの処理手順を第３図のフローチャートに基づいて説明する。

まずサムチェックを行なう前に、ＣＰＵ２０は、イメージリーダー１４から入力されたイメージデータをイメージメモリ１１のイメージ領域へ蓄積させると共に、ハードディスク１２に記憶されたフォントファイルを読み出し、イメージメモリ１１のフォント領域へ蓄積させる。そして、このイメージメモリ１１内でイメージとフォントの合成・加工を行なう。

次に、ＣＰＵ２０は、イメージメモリ１１のバンクｊに初期値１を与える（ステップ１０１）。これによりバンク１が指定される。続いて、イメージメモリ１１の０バンク目のアドレスｊに格納されたＳＵＭ－ｊ（チェックサム）を取り出し、ＳＵＭとする（ステップ１０２）。そして、イメージメモリ１１のバンクｊのアドレスｉを０にクリアする（ステップ１０３）。次に、前記ＳＵＭ（ＳＵＭ－ｊ）とイメージメモリ１１のバンクｊのアドレスｉ（ここではアドレス０）のデータを加算し、その値をさらにＳＵＭに加える（ステップ１０４）。次に、バンクｊのアドレスｉに１を加え、これを新たなアドレスｉとしてＳＵＭ＋ＩＭ（ｊ，ｉ）を行なう（ステップ１０５）。すなわち、次のアドレス１のデータがＳＵＭに加えられることになる。そして、上記処理を順次繰り返し、ｉ＝４０９６となったときに（ステップ１０６）、ＳＵＭ←ＳＵＭ＋ＩＭ〔ｊ，０〕＋ＩＭ〔ｊ，１〕＋…＋ＩＭ〔ｊ，４０９５〕におけるＳＵＭが０となるかどうかを判断する（ステップ１０７）。ここで、ＳＵＭ＝０であれば、そのバンクのデータを正常と判断し、バンクｊに１を加え、新たなバンクｊについて同様にＳＵＭ＋

ＩＭ（ｊ．ｉ）を行なう（ステップ１０８）。そして、上記処理を各バンクについて順次繰り返し、バンクｊｍａｘ（この例では１１）となったかどうかを判断し（ステップ１０９）、ｊｍａｘであればエラーなしと判定して処理を終了する。また、ｊｍａｘでなければｊｍａｘとなるまで処理を繰り返す。

一方、ステップ１０７においてＳＵＭ＝０とならない場合は、バンクｊのサムチェックエラーと判定し（ステップ１１０）、エラー処理として、外部にエラー表示を行なうと共に、イメージメモリ１１からプリンタ１３へのイメージデータの送出を中断する。そして、装置をストップさせ、イメージメモリ内のイメジデータをクリヤし、フォントファイルの再ロードを行なう（ステップ１１１）。

このように、各バンクごとにアドレス０～４０９５までの全てのデータを順次加算し、これをさらにチェックサムと合計してサムチェックを行なうことにより、各バンクごとにデータエラーの有無を検出することができる。したがって、使用者はデータエラーが検出されたときは、該当するバンクのデータだけを見直せばよいので、エラーチェックを容易に行なうことができる。

なお、上記実施例では、チェックサムを論理和＝：０で説明したが、排他的論理和（ＥｘＯＲ）、ＣＲＣ等の方法を用いてもよい。

また、この実施例では、フォントファイルの先頭（０バンク目）に全バンク分のチェックサムテーブルが設けられているが、各バンクの先頭もしくは終りにそのバンク分のチェックサムを設けてもよい。

さらに、この実施例では、フォントファイルを４０９６ワード単位のバンクに分割しているが、このワード数は増減させてもよいし、フォントファイルが複数ある場合は１つのフォントファイルに１つのチェックサムを設定し、ファイルごとにデータエラーの有無をチェックするようにしてもよい。

〔発明の効果〕

以上説明したように、この発明によれば、フォントファイルにチェックサムを設け、このチェックサムに基づいてサムチェックを行ない、フォントファイルのデータエラーの有無を検出するようりにしたため、操作ミス等によりフォントファイルを破壊してしまっても、データエラーの発生した箇所を簡単に知ることができるので、エラーチェックを容易に行なうことが可能となる。

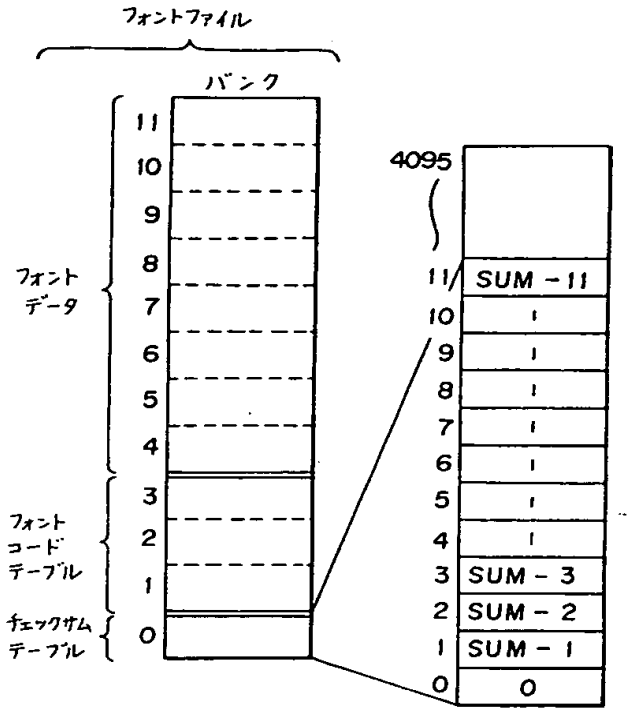
また、フォントファイルをハードディスク等の記憶装置からイメージメモリにロードする際に、このサムチェックを行なうことにより、ファイル時又はロード時のフォントファイルのデータエラーを検出することができる。

４．図面の簡単な説明

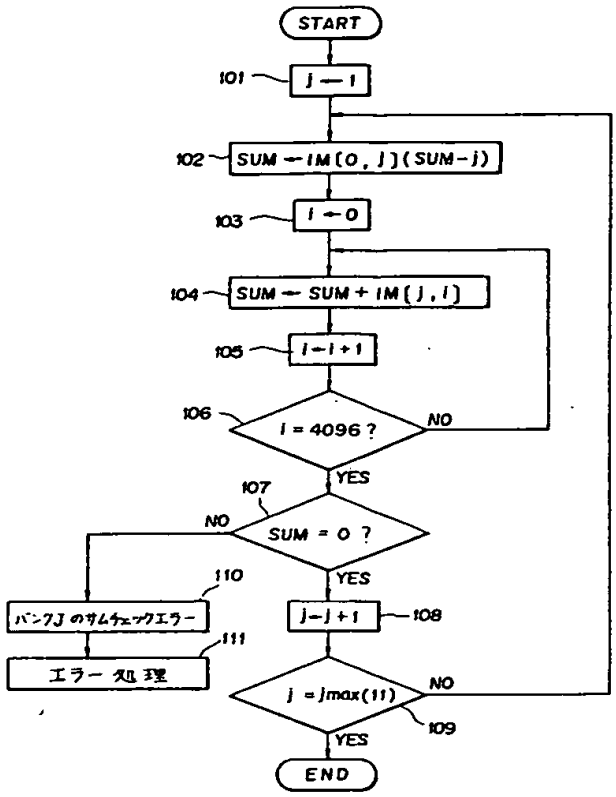
第１図はこの発明に係るイメージ処理装置のフォントファイルの一実施例を示す概念図、第２図はこの発明に係るイメージ処理装置の一実施例を示す構成ブロック図、第３図はサムチェックの処理手順を示すフローチャート、第４図は従来のイメージ処理装置の構成ブロック図である。

１１…イメージメモリ、１２…ハードディスク、１３…プリンタ、１４…イメージリーダー、１８…ＲＯＭ、１９…ＲＡＭ、２０…ＣＰＵ（中央処理装置）。

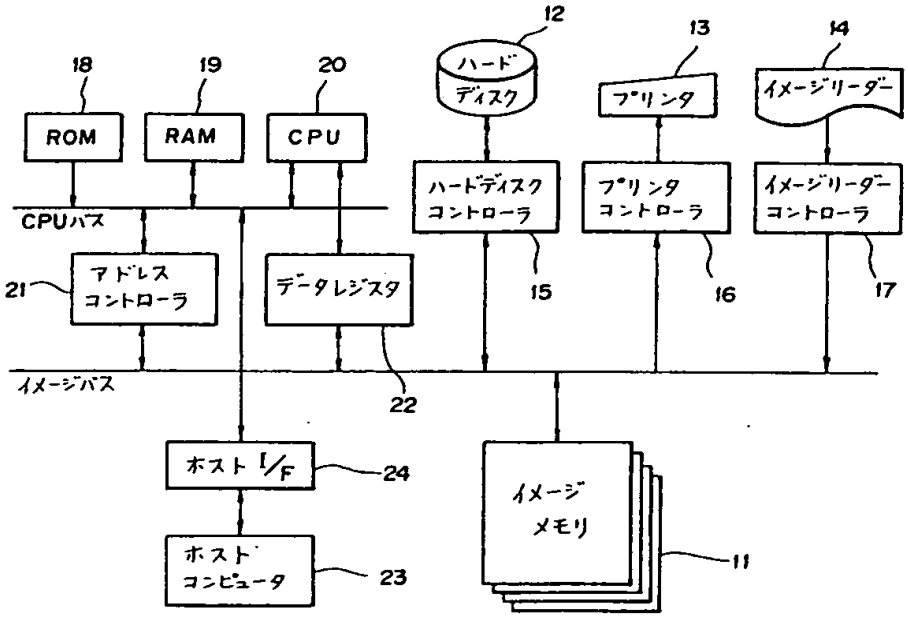
出願人代理人　木　村　高　久


フォントファイル
バンク
フォントデータ
フォントコードテーブル
チェックサムテーブル
11
10
9
8
7
6
5
4
3
2
1
0
4095
SUM - 11
SUM - 3
SUM - 2
SUM - 1
0

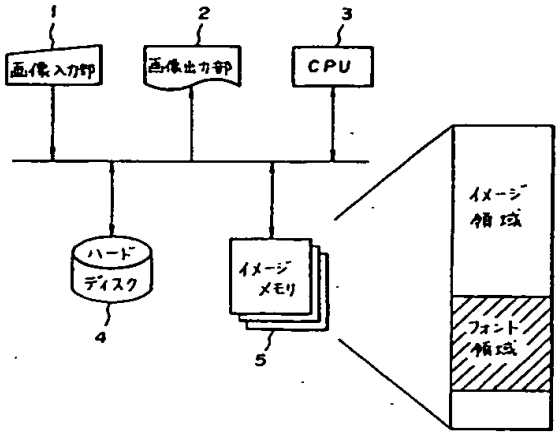
第 1 図


START
101
j ← 1
102
SUM ← IM(0, j)(SUM-j)
103
i ← 0
104
SUM ← SUM + IM(j, i)
105
i ← i + 1
106
i = 4096 ?
NO
YES
107
SUM = 0 ?
NO
YES
110
バンクJ のサムチェックエラー
111
エラー 処理
j ← j + 1
108
j = jmax(11)
NO
109
YES
END

第 3 図


18
19
20
12
13
14
ROM
RAM
CPU
ハードディスク
プリンタ
イメージリーダー
CPUバス
ハードディスクコントローラ
プリンタコントローラ
イメージリーダーコントローラ
21
アドレスコントローラ
データレジスタ
15
16
17
イメージバス
22
ホスト I/F
24
イメージメモリ
ホストコンピュータ
23
11

第 2 図



第 4 図